

Command Support CSCI

Requirements and Design Specification Review

June 19,1997

Version 1.0

1. Command Support CSCI

1.1 Command Support Introduction

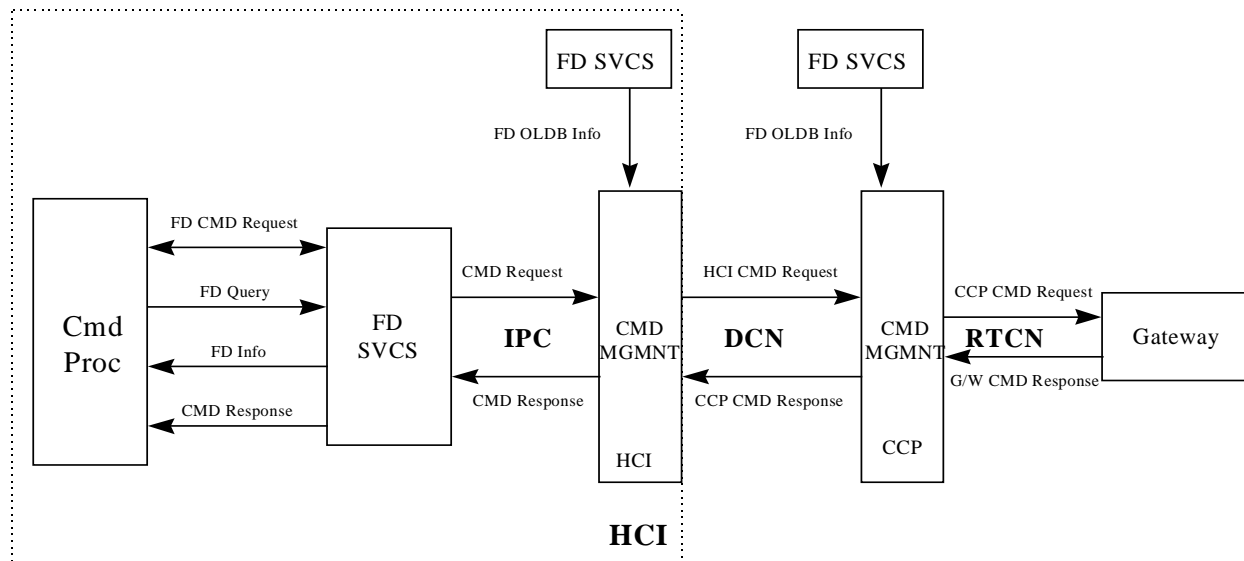
The Command Support CSCI provides users and applications the ability to command the RTPS. The Command Support CSCI consists of a command text entry user display, Command Processor, and a command authentication and routing function, Command Management.

Section 2 describes Command Processor.

Section 3 describes Command Management.

1.1.1 Command Support Overview

The user interface, Command Processor, accepts text entry input. The text is processed to form a command and Applications Services is called to create and send the command. Commands are received by the HCI resident Command Management. Command Management validates the authentication of the user and forwards the command to the CCP. A CCP resident Command Management receives commands from HCIs, performs authentication and routes the commands to the Gateways. Gateway Command Responses are sent back to the CCP Command Management and are then sent to the appropriate source.



Command Support Overview

1.2 Command Support Specifications

1.2.1 Command Support Common Ground Rules

1.2.1.1 Command Support for Redstone will not include:

- A) Prerequisite Control Logic
- B) Test Application Scripts
- C) End Item Manager

1.2.2 Command Support Functional Requirements

1.2.2.1 The Command Support CSCI shall provide the RTPS capability to issue two types of FD commands.. The commands are ***Set*** for discrete FD's and the ***Apply*** for analog FD's.

1.2.3 Command Support Performance Requirements

See Command Support CSCs for performance requirements.

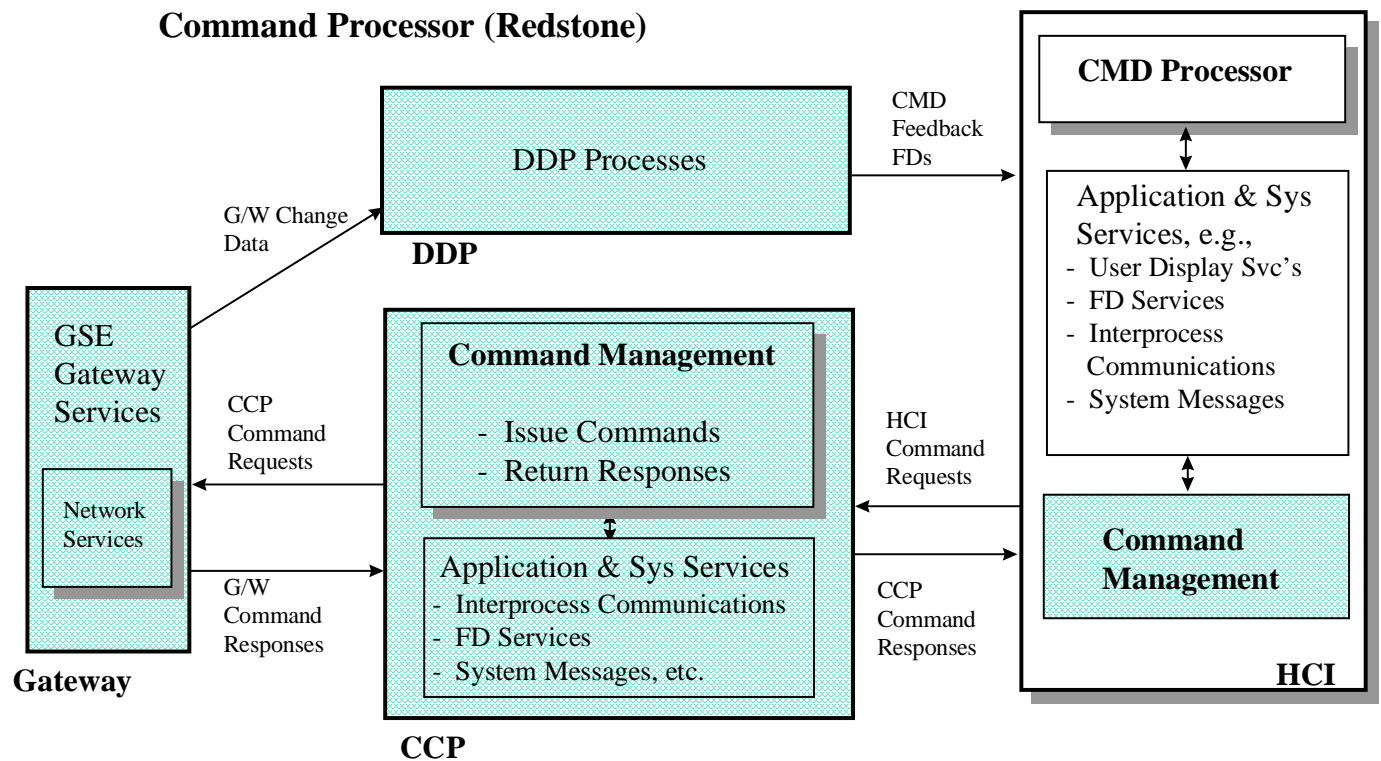
2. Command Processor CSC

2.1 Command Processor Introduction

2.1.1 Command Processor Overview

The Command Processor CSC resides in the Human Computer Interface (HCI) and provides the capability for the HCI user to issue Command Requests to end items attached to the RTPS. The Command Processor receives CCP Command Responses for each Command Request, and displays an appropriate messages on the Command Processor screen and/or issues a message to System Message Services.

The data flow of the Command Processor is as follows:



2.1.2 Command Processor Operational Description

The Command Processor provides a means to enter Command Requests to end items attached to the RTPS by a User at an HCI (workstation). The methods by which these Command Requests are entered will be prototype procedures for Redstone.

A Command Processor Command Request is made by typing a text string into a field on a User display screen. The Command Processor parses the text string to create a Command Request. Error messages are displayed indicating syntax or commanding errors. The Command Processor validates whether the Command Request is for an existing FD and is a supported command. A Command Request will be sent if the entry is processed without error and accepted by Two Step Processing.

The Command Processor receives status about Command Requests from several stages in the command route. These messages consist of Two Step Command Processing error/status messages and CCP Command Responses. These messages/responses are displayed on the Command Processor screen and are sent to System Message Services.

2.2 Command Processor Specifications

2.2.1 Command Processor Ground Rules

2.2.1.1 User Display Services will be required to support the Command Processor to:

- A) provide a modeless pop-up window for 2 step command options.
- B) provide the user the ability to cancel a window.

2.2.1.2 Services will be required to support the Command Processor to:

- A) search for a FD by FD ID
- B) search for a FD by FD Name
- C) return the following FD attributes:
 - a) FD ID
 - b) FD Name
 - c) Current value
 - d) FD type
 - e) Options (for Discrete only - Open/ Close, On/Off, etc)
 - f) Range (for Analog only - Min, Max, precision)
 - g) Routing code
 - h) Destination

2.2.1.3 System Services Interprocess Communications will be required to support the Command Processor CSC communications with the Command Management CSC.

2.2.1.4 The Command Processor will output error messages to System Message Services.

2.2.2 Command Processor Functional Requirements

2.2.2.1 The Command Processor shall provide the facility to structure and send manually typed Command Requests.

2.2.2.2 The commanded objects shall be designated by FD names.

2.2.2.3 The Command Processor Set command shall use the correct discrete definition for the FD. For example:

- a) Open/Close
- b) True/False
- c) Wet/Dry
- d) On/Off

2.2.2.4 The Command Processor Apply command shall use numeric input for analog FD's.

Software Requirements and Design Specification

2.2.2.5 The Command Processor shall receive Command Responses from the CCP concerning each Command Request. The responses shall be routed to the User's Screen and System Message Services according the table below.

	CCP Command Response	Command Processor Message	Additional Messages from Command Processor
a	Authentication Problem	Error Message	No Message
b	Command Time-out	Error Message	No Message
c	Command issued	No Message	No Message
d	Command completed	Status Message	No Message
e	Command Error (non-HIM error)	Error Message	Message to System Message Services
f	Command Error (HIM error)	Error Message	No Message

2.2.2.6 The Command Processor shall restrict invalid Command Requests

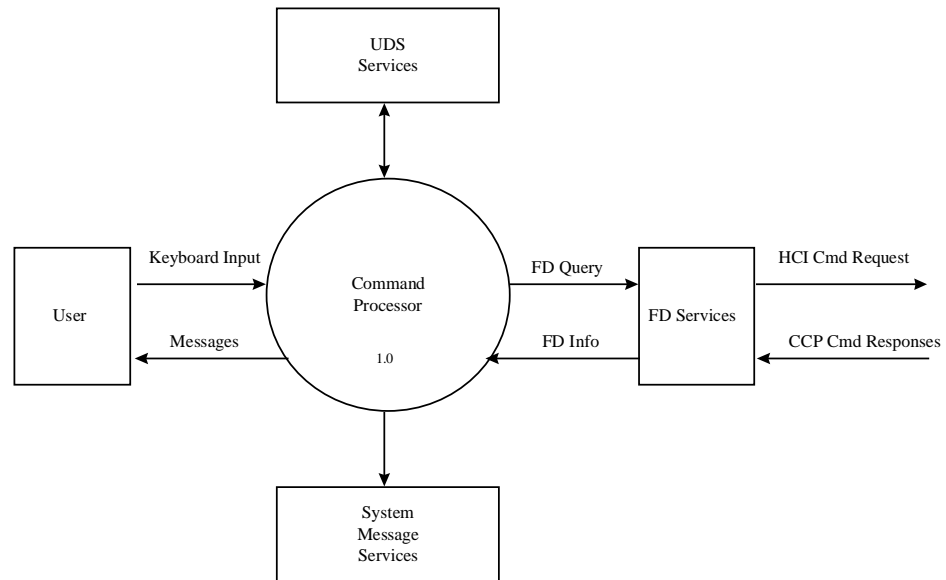
- a) The Command Processor will reject invalid FD names.
- b) The Command Processor will reject values out-of-range for an intended FD.
- c) The Command Processor will reject invalid value types for an intended FD.

2.2.2.7 The Command Processor shall support two-step Command Requests (arm First, then execute).

2.2.3 Command Processor Performance Requirements

Command Processor Performance Requirements are not defined for the Redstone Release.

2.2.4 Command Processor Interfaces Data Flow Diagram



Command Processor Overview Level 0

2.3 Command Processor Design Specification

The Command Processor presented here will be an OOD specification and implemented using the C++ language.

The Command Processor consists of a user interface window that accepts text input from the user, a text string parser, and a Response Processor.

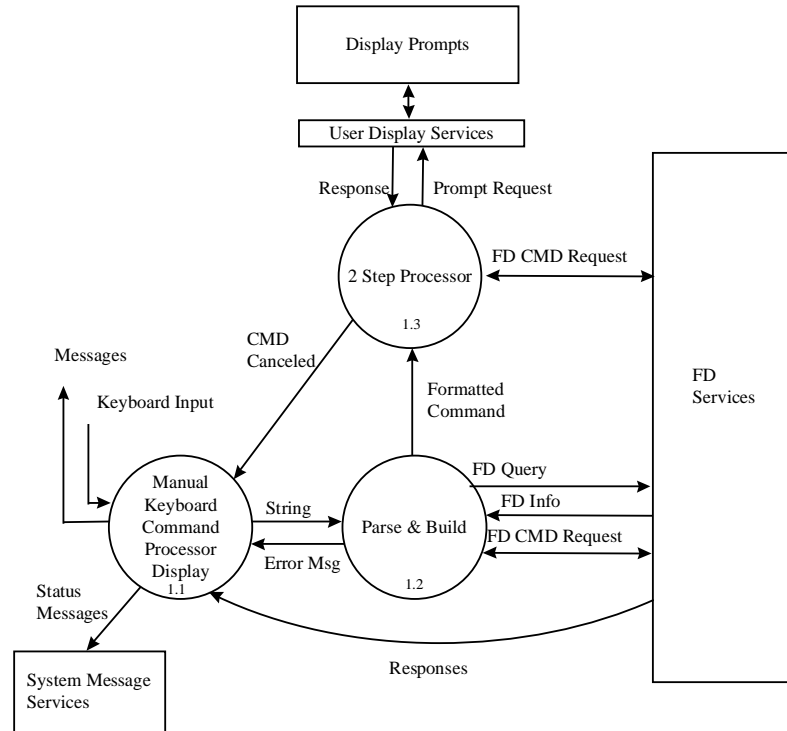
The Command Processor consists of at least the following classes:

1. Parse and Build Processor
2. Two Step Processing

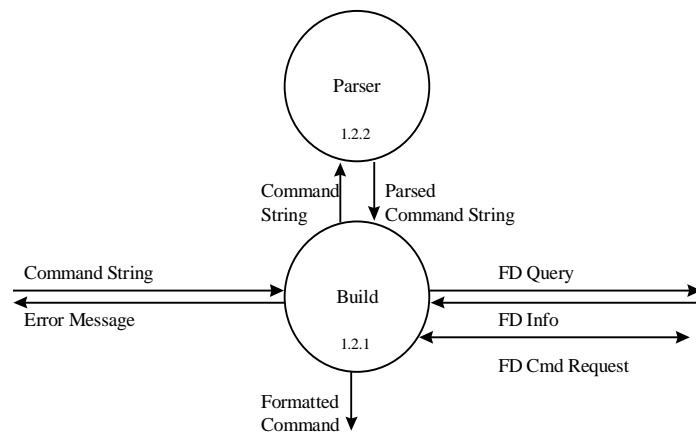
2.3.1 Command Processor Detailed Data Flow

2.3.1.1 Command Processor Detailed Data Flow

Software Requirements and Design Specification



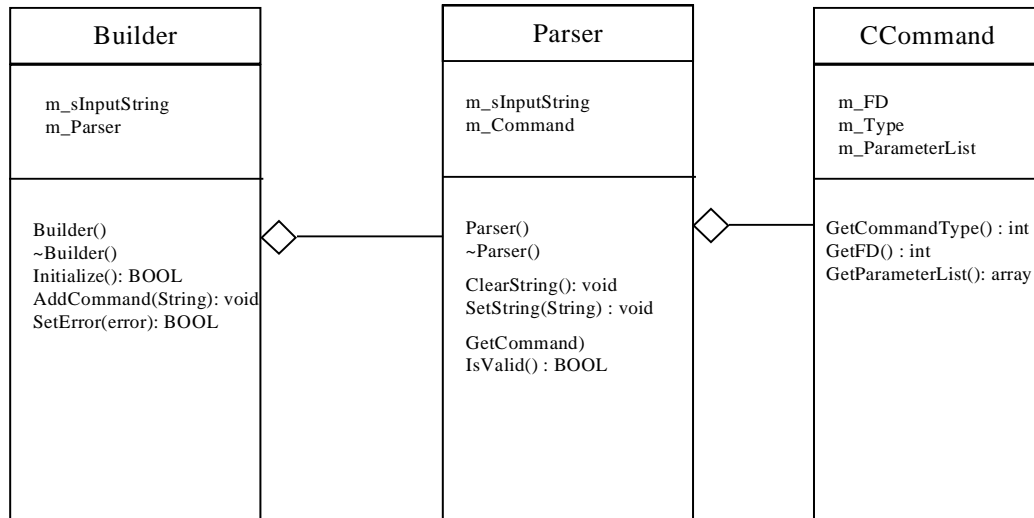
Command Processing Detailed Data Flow Level 1



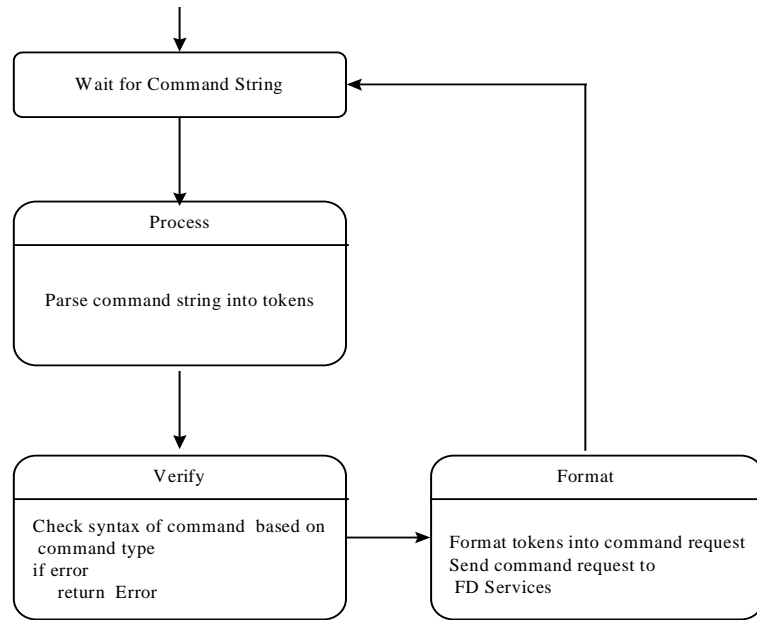
Parser Detailed Data Flow Level 1.2

2.3.1.2 Command Processor Class Diagrams

Software Requirements and Design Specification



Class diagram for Parser/Builder



State Transition Diagram for Parser/Builder

2.3.2 Command Processor External Interfaces

2.3.2.1 Command Processor Message Formats

The Command Processor will issue the following responses to the Command Processor window when processing a string input or a Command Response:

Software Requirements and Design Specification

CMD PROC Message Number	CPRO Message Number	Message Text
1	314	Apply Failed - Invalid Value to Apply
2	413	Apply Successful FD Was R1 Is R2 R3
3	316	Set Failed - Invalid State to Set
4	415	Set Successful FD Was I1 Is I2
5		Invalid Command A1
6		CMD Invalid FD A1
7		CMD Canceled in 2 Step Processing - FD
8		Communications Failure Unable to send Command to Command Management CMD , FD , V1 , S1 *
9		Communications Failure Unable to send Command to Gateway CMD , FD , V1 , S1 *
10		Command Time-out CMD , FD , V1 *
11		HIM Error CMD , FD , V1 HIM error number = S1 **
12		Invalid source for Command Request CMD , FD , V1 *
13		Gateway Unsolicited Command Response CMD , FD , I1 , I2 *

* This error message is displayed because of a error response in the C-to-C. Command Management has issued a System Message.

** This error message is displayed because of a error response in the C-to-C. The Gateway has issued a System Message.

Inserts:

Insert ID	Insert Value
A1	Text String
CMD	Command Name
FD	FD ID
I1	Previous Value
I2	Current Value
R1	Previous Value
R2	Current Value
R3	Units
S1	Status/error code
V1	Value sent

Message Number = 1

Severity = Warning

Apply Failed - Invalid Value to Apply

Help Information Content:

The value in the command string was either an invalid Apply type or out of range.

Details information:

N/A

Message Number = 2

Severity = Informational

Software Requirements and Design Specification

Apply Successful FD Was R1 Is R2 R3

FD = FD Name
R1 = Previous Value
R2 = Current Value
R3 = Units

Help Information Content:

The command has been received by the destination and executed without error.

Details information:

N/A

Message Number = 3

Severity = Warning

Set Failed - Invalid State to Set

Help Information Content:

Invalid state to set..

Details information:

N/A

Message Number = 4

Severity = Informational

Set Successful FD Was I1 Is I2

FD = FD Name
I1 = Previous Value
I2 = Current Value

Help Information Content:

The command has been received by the destination and executed without error.

Details information:

N/A

Message Number = 5

Severity = Warning

Invalid Command A1

A1 = Text String

Help Information Content:

The command is invalid. Check spelling of the command text.

Details information:

N/A

Message Number = 6

Severity = Warning

CMD Invalid FD A1

CMD = Command Name

A1 = Text String

Help Information Content:

The FD is invalid for the command. Check the spelling and/or type of the FD string.

Details information:

N/A

Message Number = 7

Severity = Warning

CMD Canceled in 2 Step Processing - FD

CMD = Command Name

FD = FD Name

Help Information Content:

The user has canceled the command in 2 Step Processing. No command issued.

Details information:

N/A

Message Number = 8

Severity = Error

Communications Failure Unable to send Command to Command Management. CMD, FD, I3, I4

CMD = Command Name

FD = FD Name

Insert #3 = Value being sent

Insert #4 = IPC Error Code returned

Help Information Content:

Command Processor was unable to issue a C-to-C. IPC returned an error code.

Details information:

Insert #4 contains the error code returned by IPC. See IPC codes.

Message Number = 9

Severity = Error

Communications Failure Unable to send Command to Gateway. CMD, FD, I3, I4

CMD = Command Name

FD = FD Name

Software Requirements and Design Specification

Insert #3 = Value being sent

Insert #4 = IPC Error Code returned

Help Information Content:

Command Processor was unable to issue a C-to-C. IPC returned an error code.

Details information:

Insert #4 contains the error code returned by IPC. See IPC codes.

Message Number = 10

Severity = Error

Command Time-out. CMD, FD, I3

CMD = Command Name

FD = FD Name

Insert #3 = Value being sent

Help Information Content:

A Command Response has not been received in the time allocated for the command to complete.

Details information:

N/A

Message Number = 11

Severity =

HIM Error. CMD, FD, I3, HIM error number = I4

CMD = Command Name

FD = FD Name

Insert #3 = Value being sent

Insert #4 = HIM Error number

Help Information Content:

The Gateway experienced a HIM error.

Details information:

Insert 4 is the HIM error code returned by the Gateway.

Message Number = 12

Severity =

Invalid source for Command Request. CMD, FD, I3

CMD = Command Name

FD = FD Name

Insert #3 = Value being sent

Help Information Content:

The console that issued the command request was not authorized to issue the command.

Software Requirements and Design Specification

Details information:

N/A

Message Number = 13

Severity =

Gateway Unsolicited Command Response. CMD, FD, I3, I4

CMD = Command Name

FD = FD Name

Insert #3 = Value being sent

Insert #4 = Value returned.

Help Information Content:

Gateway has issued an unsolicited command response. The command timed out in Command Management and this is a late response.

Details information:

N/A

The Command Processor will display on the Command Processor window and send to System Message Services the following messages when processing a Command Response.

Message Group = **CMD**

Message Number	Message Text
S1	Unauthorized command.
S2	Command Error returned from Gateway

Message Number = S1

Severity = Error

Unauthorized command. CMD, FD, I3

CMD = Command Name

FD = FD Name

Insert #3 = Value being sent

Help Information Content:

The source task does not have permission to issue the command, If the source is a user display the user has attempted to issue a command that is not enabled for the station.

Details information:

N/A

Message Number = S2

Severity = Error

Command Error returned from Gateway. CMD, FD, I3, I4

Insert #1 = Command Type (Set, Apply)

Insert #2 = FD Name

Insert #3 = Value being sent

Insert #4 = Error Code

Help Information Content:

The Gateway determined a fault during the execution of the command.

Details information:

Insert #4 is the error code returned by the Gateway.

2.3.2.2 Command Processor Display Formats

Command Processor Prototype Display 1

Two Step Prompt

2.3.2.3 Command Processor Input Formats

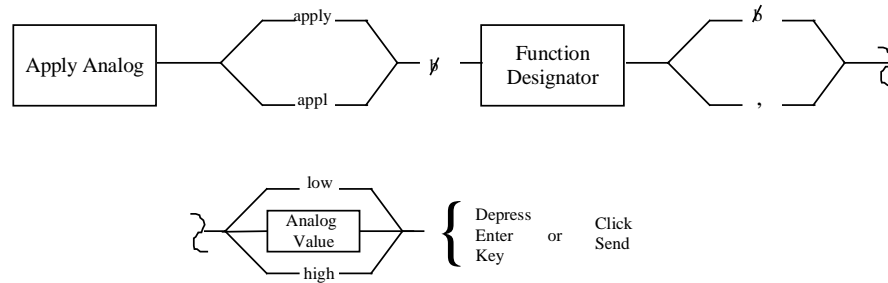
The Command Processor shall support the following language formats:

General rules:

- A. Commands shall be case insensitive. Lower and upper case letters are completely interchangeable. For example: a lower case “a” is equal to a upper case “A”.
- B. A blank delimiter is equal to one or more blank characters.
- C. A comma delimiter is equal to exactly one comma.

Software Requirements and Design Specification

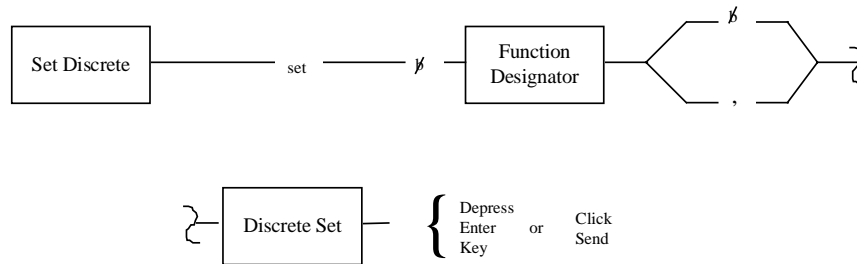
1. APPLY ANALOG:



Apply Analog rules:

- A. Apply is a two step command.
- B. Valid FD types are AS (Analog Stimulus) or PA (Pseudo Analog).
- C. The use of the word LOW or HIGH will cause the lowest or highest Engineering Unit for the FD possible to be used.

2. SET DISCRETE:



Set Discrete rules:

- A. Set is a two step command.
- B. Valid FD types are DS (Discrete Stimulus) or PD (Pseudo Discrete).

2.3.2.4 Command Processor Recorded Data

N/A

2.3.2.5 Command Processor Printer Formats

N/A

2.3.2.6 Command Processor Interprocess Communications (C-to-C Communications)

The Command Processor sends Command Requests and receives Command Responses to/from the Command Manager using Application Services, FD Services API.

The Command Processor sends System Messages using the System Message Services API.

2.3.2.7 Command Processor External Interface Calls (e.g., API Calling Formats)

The Command Processor uses the following APIs:

1) Application Services:

FD Services:

Refer to FD Services Interface Description Document, Document Number 84KXXXX.

2) System Services:

System Message Services:

TBD.

2.3.2.8 Command Processor Table Formats

N/A

2.3.3 Command Processor Test Plan

Command Processor Test Cases:

1. Initialization Command Display
2. Exit Command Display
3. Bad command input (not Set or Apply)
4. Apply variations
 - A. Good and bad syntax
 - B. Out of range
 - C. Two step processing
5. Set variations
 - A. Good and bad syntax
 - B. Out of range
 - C. Two step processing
6. Command Authority.
7. Test time-out processing
8. Command Response processing Error C-To-C

The various configurations that the Command Processor will be tested in is discussed in detail in the User Commanding Phase 1 Thread Presentation.

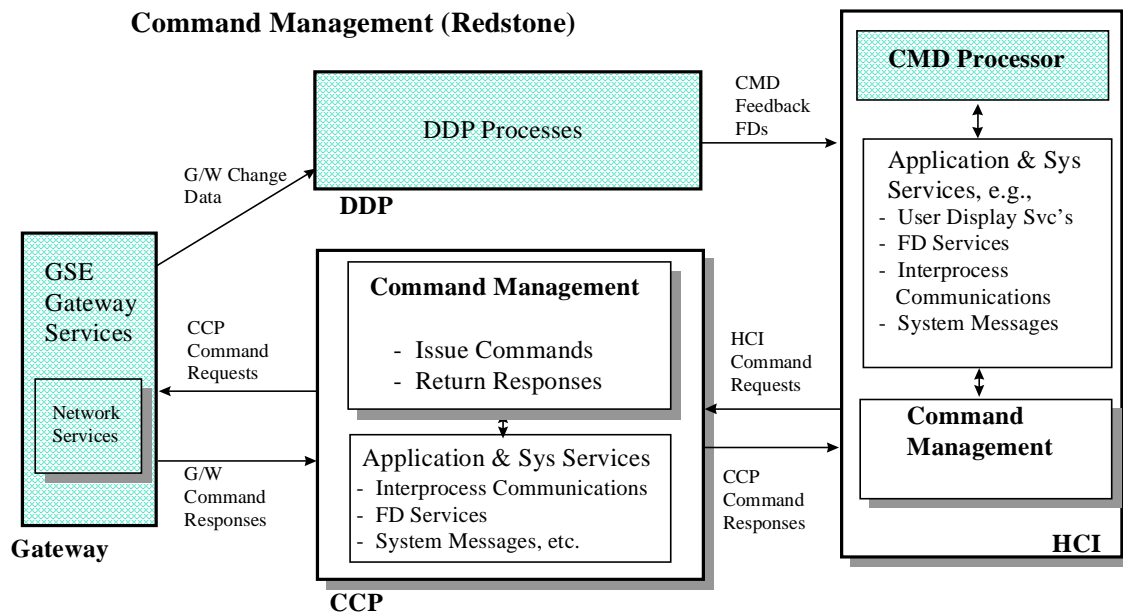
3.

Command Management CSC

3.1 Command Management Introduction

3.1.1 Command Management Overview

Command Management is a service that provides applications the ability to issue Command Requests and receive Command Responses. Command Management monitors the Command Requests for time out conditions allowing applications to perform other tasks.



3.1.2 Command Management Operational Description

Command Management receives Command Requests, performs some authentication and stores information about the Command Requests into a transaction table. The Command Requests are then forwarded to their destination, and the transaction table is monitored in case the Command Request should time out. Command Responses are received from the destination(s) and are matched up with their entry in the transaction table to be forwarded back to the requesting process. If a Command Request should time out, a Command Response will be generated with the appropriate error code.

3.2 Command Management Specifications

3.2.1 Command Management Ground Rules

Software Requirements and Design Specification

- 3.2.1.1 The Command Management will be started and terminated by the System Control CSCI.
- 3.2.1.2 [The Command Management will be initialized on the HCI and CCP platforms.](#)
- 3.2.1.3 [The Command Management , when initialized, will configure itself to perform certain levels of command authentication depending on which platform it resides on.](#)
- 2.2.3.1 System Services Interprocess Communications will be required to support the Command Management to:
 - A) provide communications between applications residing on the same platform.
 - B) provide communications between applications residing on different platforms. Inter-application Communications will make access to different communications services transparent.
- 3.2.1.4 All Command Requests from the Command Management must respond with one of the following Command Responses:
 - A) Error Response.
 - B) Successful Response
- 3.2.1.5 Command Responses to the Command Management must include a Packet Payload Header for each response, [formatted in accordance with the Real Time Processing System Packet Payload ICD, Document Number 84K00351](#) -which includes the following data fields:
 - A) Payload Type
 - B) Flags (Record, Response expected, etc.)
 - C) Number of bytes in Response Payload
 - D) Place/Flow Number (received in Command)
 - E) Logical Source of Sender (received in Command)
 - F) Transaction ID (received in Command)
 - G) Completion Code
 - H) Time
- 3.2.1.6 FD Command Responses to the Command Management must include a Packet Payload Body, [formatted in accordance with the Real Time Processing System Packet Payload ICD](#) which includes the following data fields:
 - A) FD ID
 - B) FD Commanded Value [\(represented in Engineering Units.\)](#)
 - C) FD Value returned from the HIM [\(represented in Engineering Units.\)](#)
 - D) [FD Commanded Value \(represented in Raw Counts\)](#)
 - E) [FD Value returned from the HIM \(represented in Raw Counts\)](#)
- 3.2.1.7 The Command Management will output error and status messages to System Message Services.

3.2.2 Command Management Functional Requirements

- 3.2.2.1 The Command Management shall initialize itself and wait for Command Requests and Command Responses when Command Management is started by the System Control CSCI.
- 3.2.2.2 [The Command Management, on the HCI platform, shall receive Command Requests from Applications using FD Services.](#)
- 3.2.2.3 [The Command Management, on the HCI platform, shall format an HCI Command Request for each Command Request received.](#)
- 3.2.2.4 The Command Management, [on the CCP platform](#), shall receive HCI Command Request messages.
- 3.2.2.5 The Command Management, [on the CCP platform](#), shall verify (Authenticate) that the HCI Command Request is issued from a configured HCI that is part of the Test Set.
- 3.2.2.6 The Command Management, [on the CCP platform](#), shall format an CCP Command Request for each HCI Command Request received.
- 3.2.2.7 The Command Management shall create a [Packet Payload](#)User Header, formatted in accordance with the Real Time Processing System Packet Payload ICD, for each Command Request which includes the following data fields:
 - A) Payload Type
 - B) Flags (Record, Response expected, etc.)
 - C) Number of bytes in Request Payload
 - D) Place/Flow Number
 - E) Logical Source of Sender

Software Requirements and Design Specification

- F) Routing ID (Command Type)
- G) Request ID (Command Sub-type)
- H) Transaction Number for Route
- I) Time

3.2.2.8 The Command Management software shall create a Packet Payload Body, [formatted in accordance with the Real Time Processing System Packet Payload ICD](#), for each Command Request which includes the following data fields:

- A) FD ID
- B) FD value (discrete or analog depending on FD type) (Analog values shall be represented in Engineering Units.)

3.2.2.9 The Command Management shall track all Command Requests for response matching.

3.2.2.10 The Command Management shall keep a TimeOut value for each Command Request sent.

- A) The Command Management shall set the TimeOut value to one of the following system default values:

Command Management Timeout Table	
Location of CM	Timeout
HCI	120 msec
CCP to GSE G/W	10 msec

- B) The Command Management shall issue an Error Response when a Command Request's TimeOut is exceeded.

3.2.2.11 The Command Management shall accept the following Command Responses:

- A) Error Response.
- B) Successful Response.

3.2.2.12 The Command Management shall accept Unsolicited Command Responses. Unsolicited Command Responses include those responses received for command requests that have timed out.

- A) The Command Management shall send a message to System Message Services for each Unsolicited Command Response received.
- B) The Command Management shall attempt to route the Unsolicited Command Response back to the commanding source of the message.

3.2.2.13 The Command Management shall match the Command Responses to the stored Command Requests in the transaction table.

3.2.2.14 The Command Management shall forward Command Responses containing a status or error message to the originating process.

	Response Message (RM)	RM Returned	System Message
a	Invalid Source (Flow) for Command Request	Y	Y
b	GW Command (Error) Response	Y	N*
c	GW Command (Successful) Response	Y	N
d	CMD MGMT Generated Gateway Busy	Y	Y
e	CMD MGMT Generated Command Issued	Y	N
f	CMD MGMT Generated Command Time Out Error Command Response	Y	Y
g	GW Unsolicited Command Responses (error)	Y**	Y

* G/W will issue HIM error messages to System Message Services.

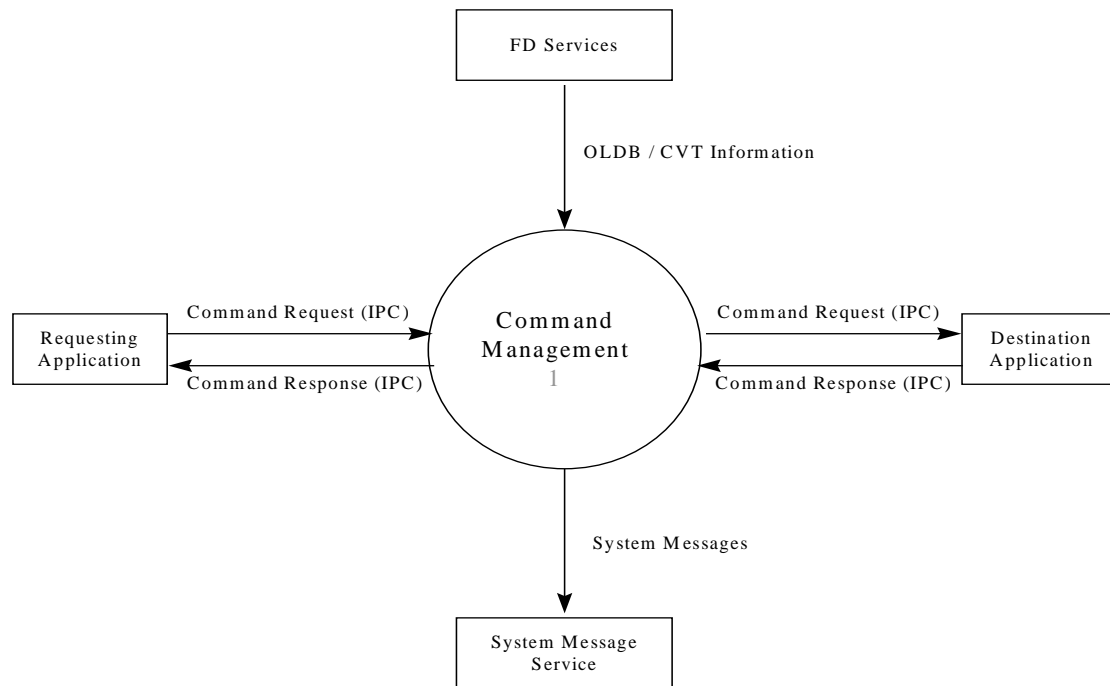
** Issuing source from Packet Header.

3.2.3 Command Management Performance Requirements

Command Management performance requirements are defined below:

CLCS System Level Specification paragraph 2.2.2.2.2	Requirement.
	RTPS shall be able to support full Uplink command rates on the following links: GSE - 500/second.

3.2.4 Command Management Interfaces Data Flow Diagram



Level 0 Data Flow Diagram for Command Management

3.3 Command Management Design Specification

The Command Management presented here will be a OOD specification and implemented using the C++ language.

Command Management will consist of at least the following classes:

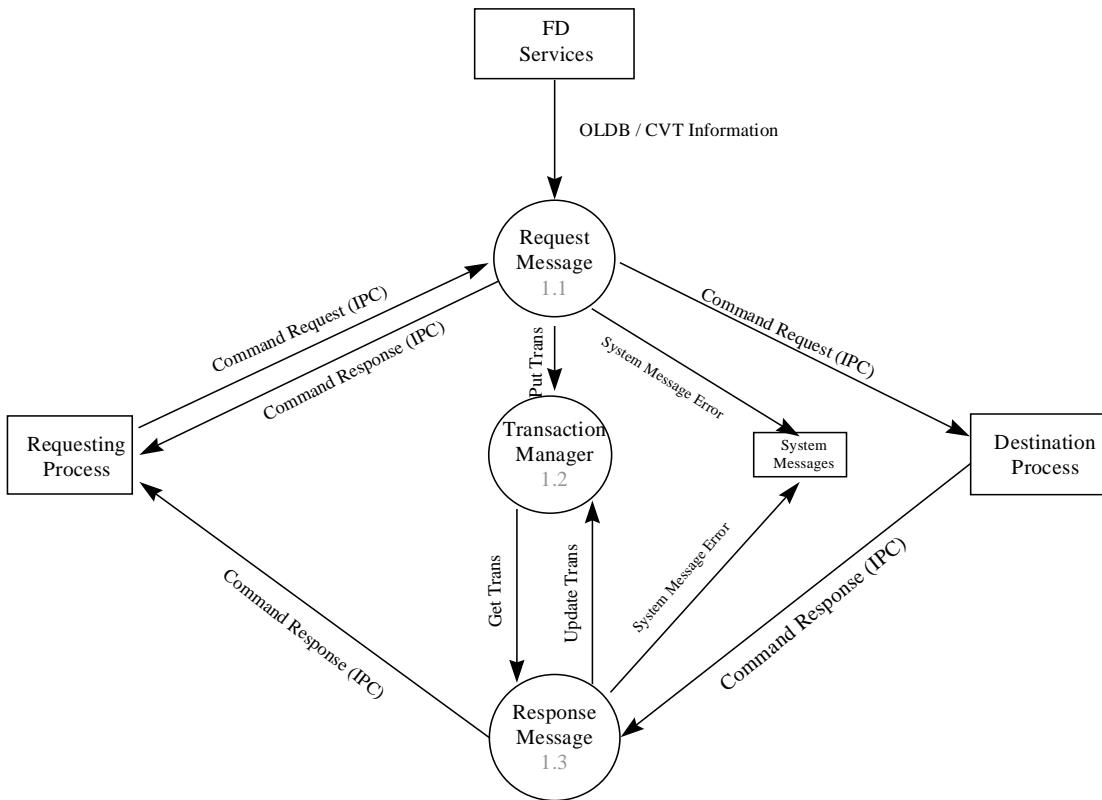
1. Command Request Processor
2. Message Authenticator
3. Command Response Processor
4. Transaction Manager.

Command Management will be started by the System Control CSCI and will monitor Command Request/Responses sent from applications.

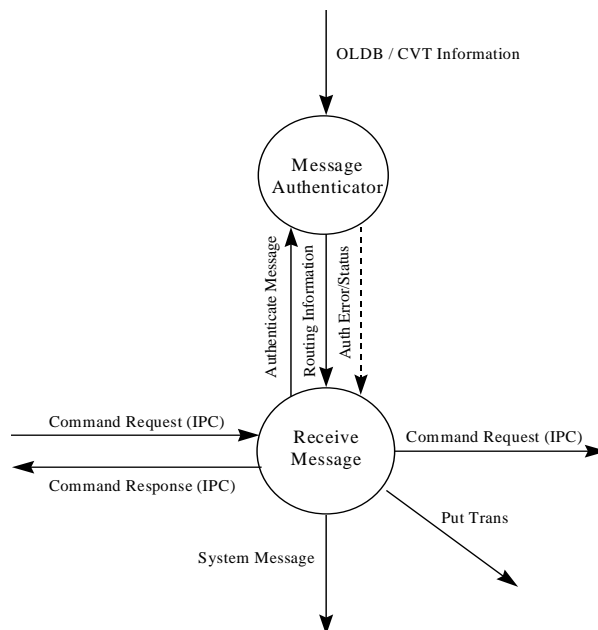
3.3.1 Command Management Detailed Data Flow

3.3.1.1 Command Management Detailed Data Flow

Software Requirements and Design Specification

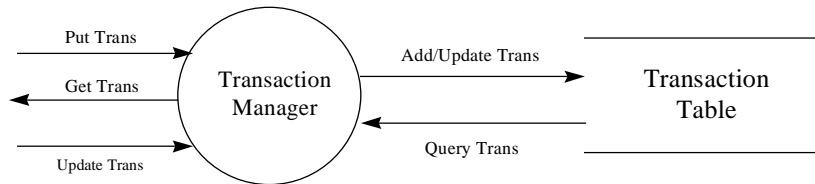


Level 1 Data Flow Diagram for Command Management

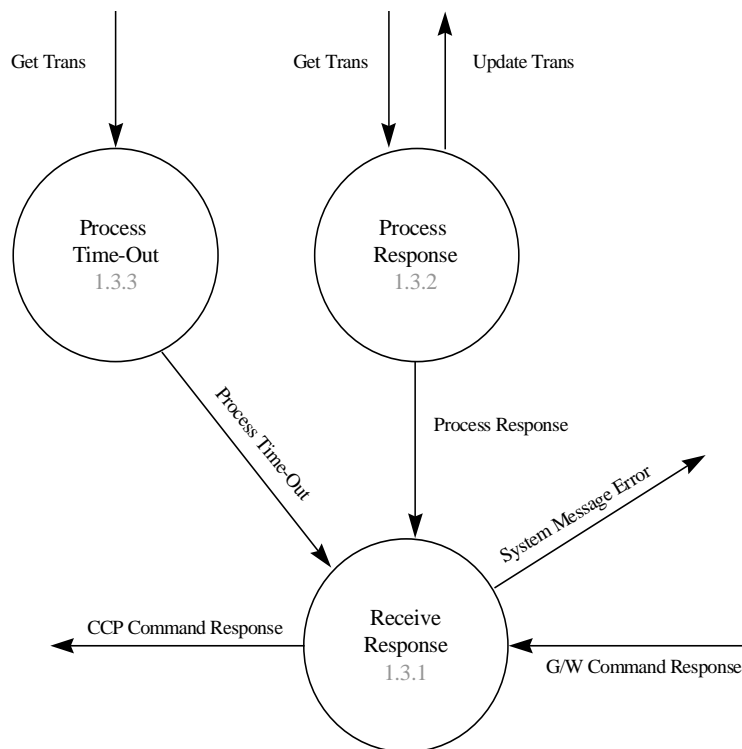


Level 1.1 Data Flow of Request Message

Software Requirements and Design Specification

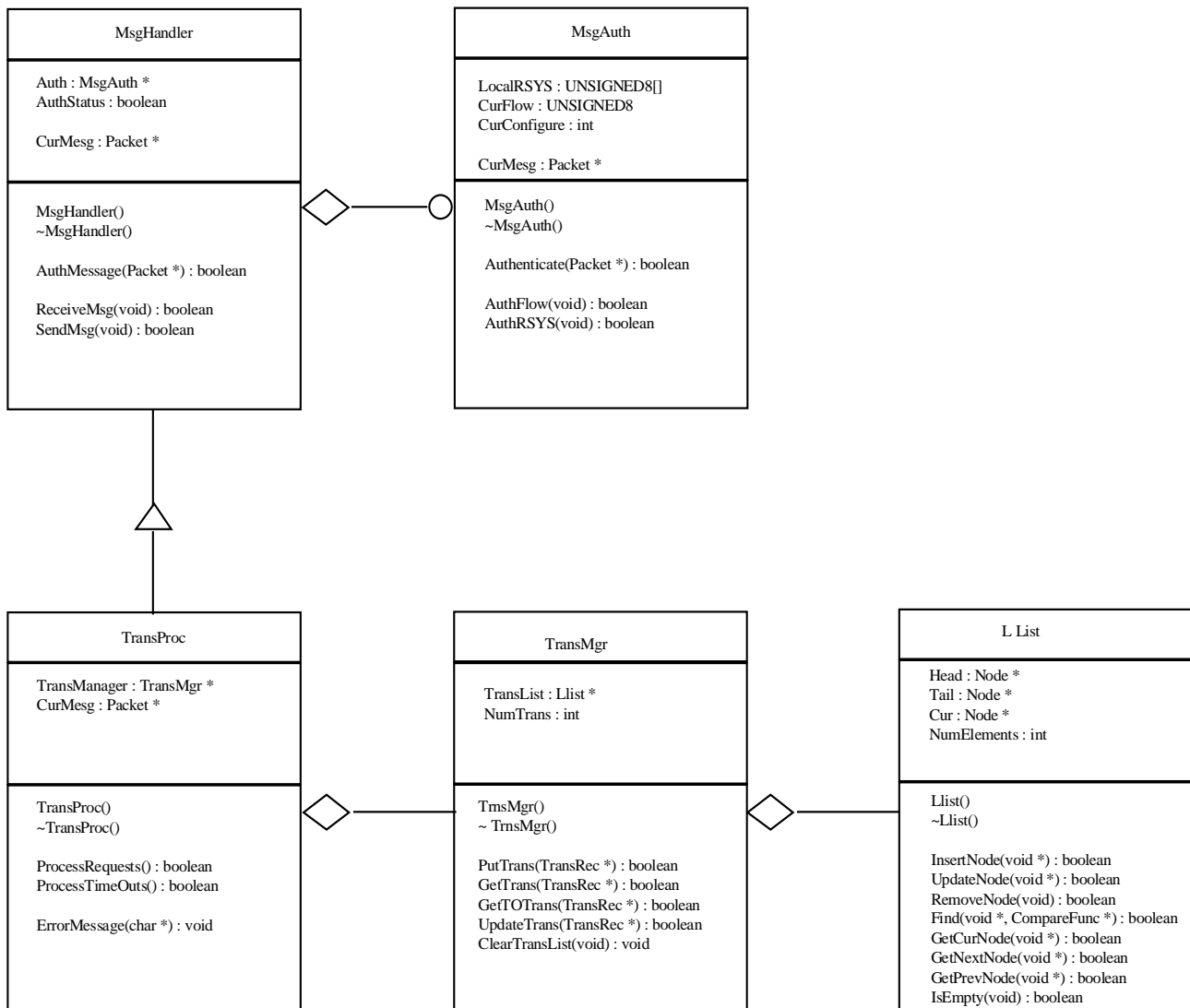


Level 1.2 Data Flow Diagram of Transaction Manager

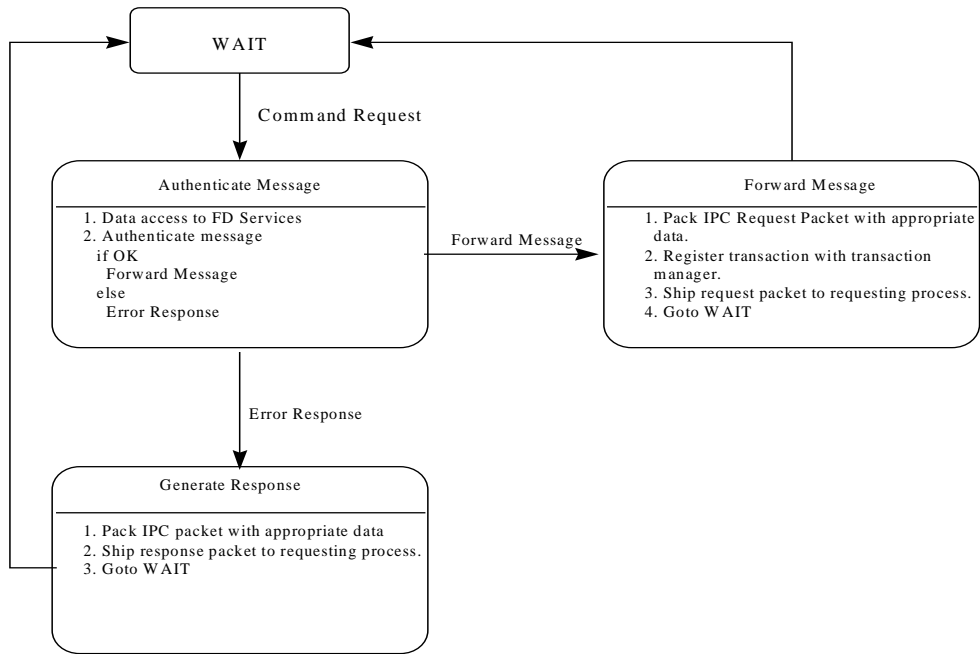


Level 1.3 Data Flow Diagram of Response Message

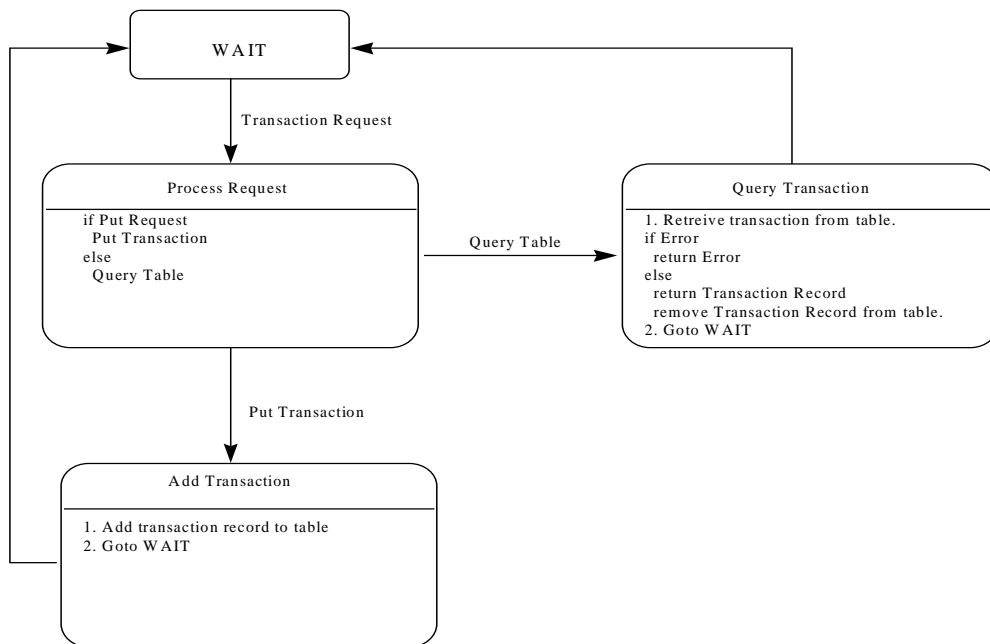
3.3.1.2 Command Management Class Diagrams



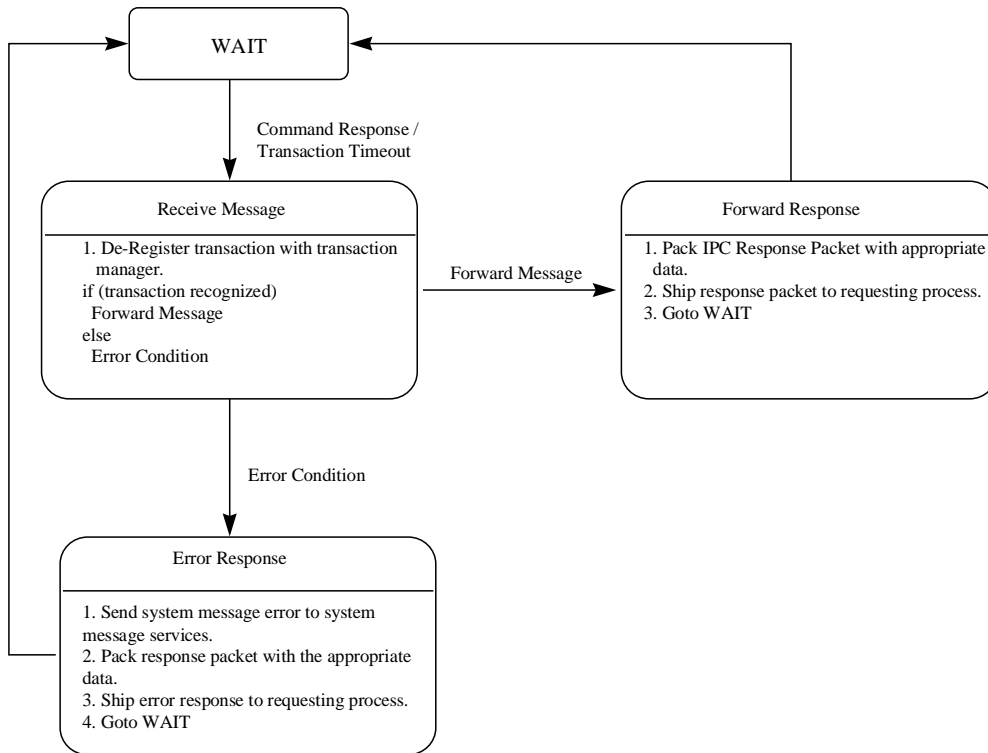
Class Diagram for Command Management



State Transition Diagram for Request Message Process



State Transition Diagram for Transaction Manager Process



State Transition Diagram for Response Message Process

3.3.2 Command Management External Interfaces

3.3.2.1 Command Management Message Formats

Command Management belongs to the Command Support CSCI Message Group.

The Command Management sends the following system messages using System Message Services:

Message Number	Message Text
1	Communications Failure: Unable to send Command Requests
2	Communications Failure: Unable to send Command Response
3	Authorization Error : Invalid Source (RSYS) for Command Request
4	Authorization Error: Command Request received from invalid Test Configuration
5	CMD MGMT Generated : Command Time-out
6	CMD MGMT Generated: Unsolicited Command Response
7	CMD MGMT Generated: Could Not Access OLDB Information
TBD	TBD

Software Requirements and Design Specification

Message Number: 1

Message Group: CMD

Severity: Error

CMD Command Management #Insert1#, Unable to send Command Request to #Insert2#.

Insert1 = ASCII string representing CMD MGMT Configuration (i.e. HCI or CCP).

Insert2 = ASCII string representing Logical ID of Destination for Command Request.

Help Information Content:

While processing Command Request, Command Management could not successfully send a Command Request to the Logical Destination.

Detailed Information:

Refer to error conditions from Event Services.

Message Number: 2

Message Group: CMD

Severity: Error

CMD Command Management #Insert1#, Unable to send Command Response to #Insert2#.

Insert1 = ASCII string representing CMD MGMT Configuration (i.e. HCI or CCP).

Insert2 = ASCII string representing Logical ID of Source for Command Request.

Help Information Content:

While processing Command Response, Command Management could not successfully send a Command Response to the Logical Source.

Detailed Information:

Refer to error conditions from Event Services.

Message Number: 3

Message Group: CMD

Severity: Warning

CMD Command Management #Insert1#, received Command Request from Invalid RSYS: #Insert2# for CMD MGMT RSYS: #Insert3#.

Insert1 = ASCII string representing CMD MGMT Configuration (i.e. HCI or CCP).

Insert2 = ASCII string representing Logical RSYS of the Command Request.

Insert3 = ASCII string representing Logical RSYS of the Command Management.

Help Information Content:

While processing Command Request, Command Management determined that the Command Request was from an Invalid RSYS.

Detailed Information:

TBD.

Message Number: 4

Message Group: CMD

Severity: Warning

CMD Command Management #Insert1#, received Command Request that does not belong to Current Test Configuration: #Insert2#, Test Configuration Received: #Insert3#.

Software Requirements and Design Specification

Insert1 = ASCII string representing CMD MGMT Configuration (i.e. HCI or CCP).

Insert2 = ASCII string representing Test Configuration of the Command Request.

Insert2 = ASCII string representing Test Configuration of the Command Management.

Help Information Content:

While processing Command Request, Command Management determined that the Command Request was not valid for the current Test Configuration.

Detailed Information:

TBD.

Message Number: 5

Message Group: CMD

Severity: Error

CMD Command Management #Insert1#, Command Request Timed Out for Command Request sent To: #Insert2#.

Insert1 = ASCII string representing CMD MGMT Configuration (i.e. HCI or CCP).

Insert2 = ASCII string representing Logical ID of Destination of Request.

Help Information Content:

While processing Command Request, Command Management either received a Time Out Response from the destination, or Request Message Timed Out.

Detailed Information:

Check to see if System Message sent for Communication Failure.

Message Number: 6

Message Group: CMD

Severity: Error

CMD Command Management #Insert1#, received an Unsolicited Command Response From Master: #Insert2# For: #Insert3#.

Insert1 = ASCII string representing CMD MGMT Configuration (i.e. HCI or CCP).

Insert2 = ASCII string representing Logical ID for Master.

Insert3 = ASCII string representing Logical ID for Destination of Response.

Help Information Content:

While processing Command Responses, Command Management received a Command Response that was not registered in it's internal transaction table. Command Management forwarded the Command Response to the Logical Source obtained from the Message Header.

Detailed Information:

TBD.

Message Number: 7

Message Group: CMD

Severity: Error

CMD Command Management #Insert1# could not access OLDB Information.

Insert1 = ASCII string representing CMD MGMT Configuration (i.e. HCI or CCP).

Help Information Content:

While processing Command Request, Command Management could not access the OLDB Information from FD Services.

Detailed Information:

TBD.

3.3.2.2 Command Management Display Formats

N/A

3.3.2.3 Command Management Input Formats

N/A

3.3.2.4 Command Management Recorded Data

Command Management will set the logging flags in the IPC Packet Header for all Command Requests and Responses. All information being sent across the DCN or RTCN Networks shall be recorded.

3.3.2.5 Command Management Printer Formats

N/A

3.3.2.6 Command Management Interprocess Communications

The CMD MGMT receives HCI Command Requests and sends CCP Command Responses from/to the Command Processor CSCI. These interprocess communications are described in Document number 84K00352, CLCS HCI to Command Management Interface Description Document. CMD MGMT sends CCP Command Requests and receives Gateway Command Responses to/from the GSE Gateway. These interprocess communications are described in Document number 84K00353, Command Management to GSE Gateway Interface Description Document.

The Command Processor sends System Messages using the System Message Services API.

3.3.2.7 Command Management External Interface Calls

The Command Management uses the following APIs:

1) Application Services:

FD Services:

Refer to FD Services Interface Description Document, Document Number 84KXXXX.

2) System Services:

Interprocess Communications:

Software Requirements and Design Specification

The following calls will be used to establish communications with IPC and send / receive IPC packets.

```
(ES_Handle *) es_register ( )  
(int connect_num) es_set_pl_opt ( ES_Handle *hand, char *connect_requested)  
(int) es_receive_packet (ES_Handle *, Packet *, int connect_num, struct timeval*)  
(int) es_send_packet (ES_Handle *, Packet *, int connect_num, int packet_size,  
    Boolean, struct timeval *time_out)
```

The following commands are specific to the IPC Header. These calls will be used to assign and retrieve values stored in the header.

```
ES_GET_PL_BUFF (int data_size, Packet *, char *data_ptr)  
ES_GET_PL_BUFF (int packet_size, Packet *, char *data_ptr)  
ES_GET_PL_COMPLETION_CODE (Packet *)  
ES_GET_PL_DATA_PTR (Packet *)  
ES_GET_PL_DEST1_APP_ID (Packet *)  
ES_GET_PL_DEST1_CPU_ID (Packet *)  
ES_GET_PL_DEST1_REFDES (Packet *)  
ES_GET_PL_DEST1_RESP_SYS (Packet *)  
ES_GET_PL_DEST2_APP_ID (Packet *)  
ES_GET_PL_DEST2_CPU_ID (Packet *)  
ES_GET_PL_DEST2_REFDES (Packet *)  
ES_GET_PL_DEST2_RESP_SYS (Packet *)  
ES_GET_PL_PAYLOAD_TYPE (Packet *)  
ES_GET_PL_REQUEST_ID (Packet *)  
ES_GET_PL_RESPONSE (Packet *)  
ES_GET_PL_RESPONSE_TRAN_ID (Packet *)  
ES_GET_PL_ROUTING_CODE (Packet *)  
ES_GET_PL_SOURCE_APP_ID (Packet *)  
ES_GET_PL_SOURCE_CPU_ID (Packet *)  
ES_GET_PL_SOURCE_REFDES (Packet *)  
ES_GET_PL_SOURCE_RESP_SYS (Packet *)  
ES_GET_PL_TRAN_ID (Packet *)  
ES_SET_PL_COMPLETION_CODE (Packet *, value)  
ES_SET_PL_DATA (Packet *, char *packet_data, int data_size)  
ES_SET_PL_DEST1_APP_ID (Packet *, value)      (tbd - Redstone)  
ES_SET_PL_DEST1_CPU_ID (Packet *, value)  
ES_SET_PL_DEST1_REFDES (Packet *, value)      (post-Redstone)  
ES_SET_PL_DEST1_RESP_SYS (Packet *, value)    (post-Redstone)  
ES_SET_PL_DEST2_APP_ID (Packet *, value)      (tbd - Redstone)  
ES_SET_PL_DEST2_CPU_ID (Packet *, value)  
ES_SET_PL_DEST2_REFDES (Packet *, value)      (post-Redstone)  
ES_SET_PL_DEST2_RESP_SYS (Packet *, value)    (post-Redstone)  
ES_SET_PL_LOGGING (Packet *, Boolean, Boolean, Boolean)  
ES_SET_PL_PAYLOAD_TYPE (Packet *, value)  
ES_SET_PL_REQUEST_ID (Packet *, value)  
ES_SET_PL_RESPONSE (Packet *, Boolean)  
ES_SET_PL_RESPONSE_TRAN_ID (Packet *, value)  
ES_SET_PL_ROUTING_CODE (Packet *, value)
```

System Message Services:

TBD.

3.3.2.8 Command Management Table Formats

N/A

3.3.3 Command Management Test Plan

The CMD MGMT CIT tests the following **:

- a) Command Authentication works properly (commands come from an Application that is authorized to command the FD, and belongs to the current Flow).
- b) Command Requests are being stored properly for later matching with a Command Response.
- c) Command Responses are properly matched up with the Command Request transactions.
- d) Timed-out transactions are being handled properly (appropriate error messages sent to system message services and Error Response Packets being generated and shipped properly).
- e) Unsolicited (timed-out) Command Responses are being handled properly (appropriate error messages sent to system message services and Error Response Packets being generated and shipped properly).
- f) Timing Requirements meet the guidelines specified in the System Level Specifications.

** See the Command Thread Presentation for the CIT Minimum and CIT Configurations.